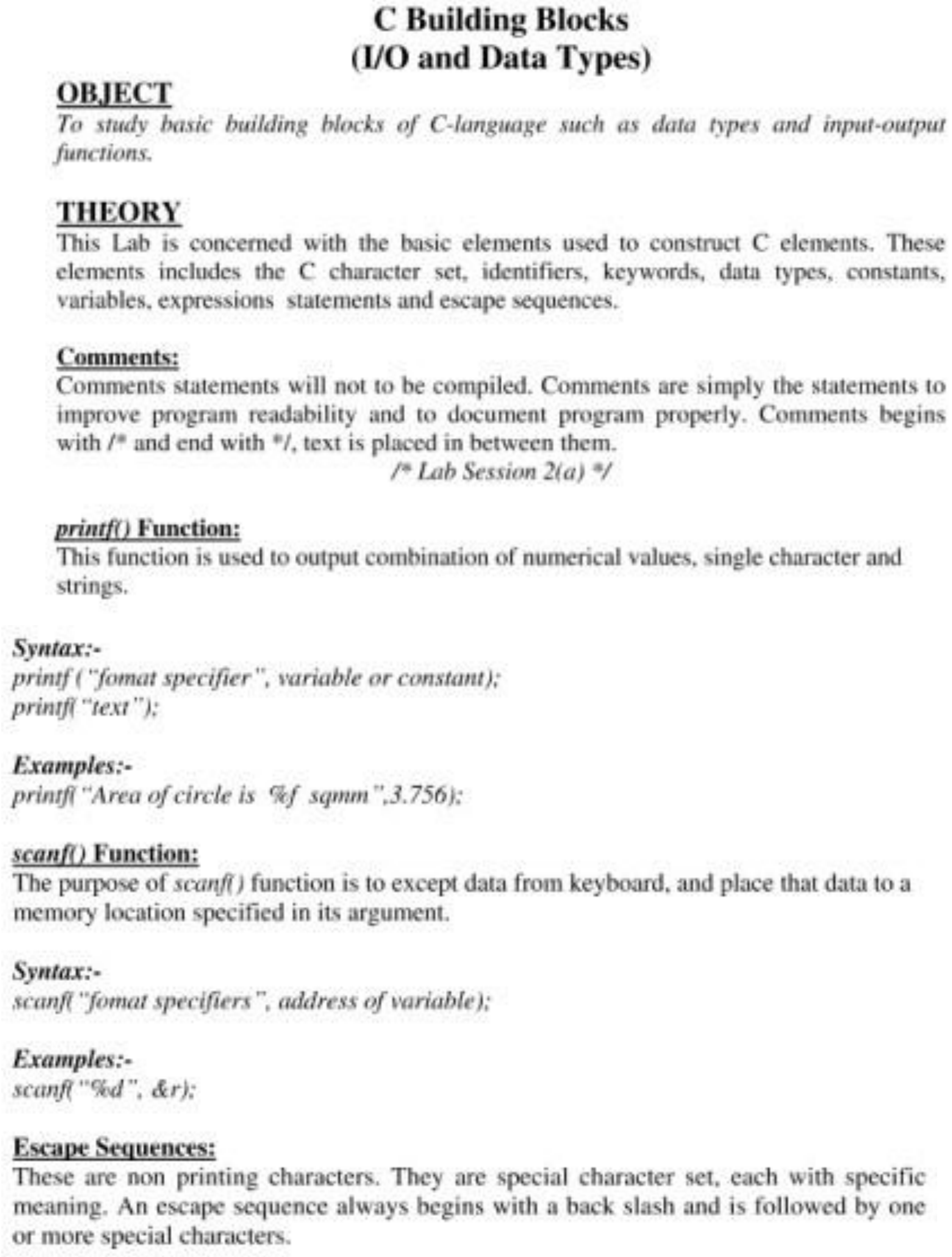
Practical # 02



C Building Blocks
(I/O and Data Types)

OBJECT

To study basic building blocks of C-language such as data types and input-output functions.

THEORY

This Lab is concerned with the basic elements used to construct C elements. These elements includes the C character set, identifiers, keywords, data types, constants, variables, expressions statements and escape sequences.

Comments:

Comments statements will not to be compiled. Comments are simply the statements to improve program readability and to document program properly. Comments begins with /* and end with */, text is placed in between them.

/ Lab Session 2(a) */*

printf() Function:

This function is used to output combination of numerical values, single character and strings.

Syntax:-

printf ("fomat specifier", variable or constant);
printf("text");

Examples:-

printf("Area of circle is %f sqmm",3.756);

scanf() Function:

The purpose of *scanf()* function is to except data from keyboard, and place that data to a memory location specified in its argument.

Syntax:-

scanf("fomat specifiers", address of variable);

Examples:-

scanf("%d", &r);

Escape Sequences:

These are non printing characters. They are special character set, each with specific meaning. An escape sequence always begins with a back slash and is followed by one or more special characters.

Table 2.1
Escape Sequences

Escape Sequence	Meaning
\n	New Line
\t	Horizontal Tab
\a	Alert(Bell)
\\	Back Slash
\"	Quotation Mark
\f	Form feed
\r	Carriage Return
\0	Null

Variables:

A variable name is a location in memory where a value can be stored for use by a program. All variables must be defined with a name and a data type in the code before they can be used in a program.

A variable name in C should be a valid identifier. An identifier is a series of characters consisting of letters, digits and underscore and does not begin with a digit. C is case sensitive i.e. area and Area can not be treated as same.

There are certain reserved words called *Keywords* that have standard, predefined meanings in C. These keywords can be used only for their intended purpose; they can't be used as programmer defined identifier.

Data Types:

C supports several different types of data, each of which may be represented differently within the computer's memory. The basic data types are listed below.

Table 2.2
Data Type and Storage Allocation

Data Type	Meaning	Bytes
char	Character	1
int	Integer	2
short	Short Integer	2
long	Long Integer	4
Unsigned	Unsigned Integer	2
Float	Floating	4
Double	Number(Decimal)	8
	Double Precision	
	Floating Point Number	

Format Specifiers:

Format specifier specifies that which type of data has to be print or read into. Following is a list of different format specifiers.

Table 2.3
Format Specifiers

Specifiers	Meaning
%c	Character
%d	Integer
%f	Float value
%e	Float value in exponential form
%u	Unsigned Integer
%x	Hexadecimal integer (unsigned)
%o	Octal value
%s	String

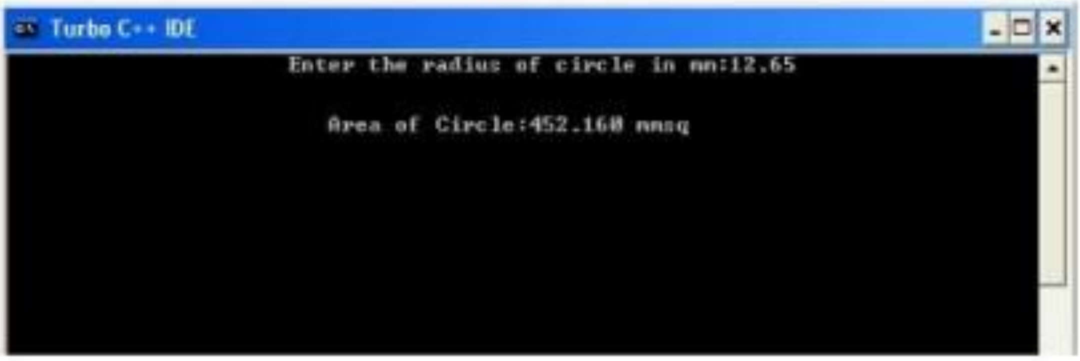
Example:

Now consider a Simple C program.

```
                        /* Lab Session 2(a) */
#include<conio.h>
#include<stdio.h>
void main (void)                /* Defining main function*/
{
clrscr();           /* Clears previous contents of screen*/
int r;             /* Declares a variable of type integer */
float area;        /* Declares a variable of type float */
float Pi=3.14;     /*Initializing a variable of type float */
printf("\t\t\tEnter the radius of circle:");        /*Output the
string on the screen*/
scanf("%d", &r); /*Stores the value on the address of variable
r*/
area=Pi*r*r;     /*Evaluating area and assigning the value to
variable area*/
printf("\n\n\t\t\tArea of Circle is :%0.3f ",area);
getch();
}
```

Even though this program is simple, it illustrates several important features of C language.

Output:



```
                       Enter the radius of circle in mm:12.65

                         Area of Circle:452.160 mmsq
```

Review Questions/ Exercise:

1. Identify and correct errors in the following statements.

a) scanf("d", value);

Ans: scanf("%d", &value);

.Added % symbol before format specifier (d).

.Added ampersand (&) before variable name (value) to pass address.

b) printf("The answer of %d+%d is"\n,x,y);

Ans: printf("The answer of %d &%d is %d"\n, x+y);

.Added %d format specifier for the answer.

.Added x+y to calculate and display the sum.

c) scanf("%d%d, &number1,number2);

Ans: scanf("%d%d", &number1, &number2);

.Added space between %d %d format specifiers for two integers.

.Added ampersand (&) before each variable's name "number1, number2" to pass addresses.

2. Write a single C statement to accomplish the following tasks. a) Prompt the user to enter an integer in the inverted commas. For example "Enter an integer" .

Ans: printf("Enter an integer: ");

.This line of code will display the message "Enter an integer: " to the user.

b) Read an integer from keyboard and store that value into a variable a.

Ans:scanf("%d", &a);

.This reads an integer from the keyboard and stores it in `a`. The `&` operator is used to provide the address of `a` to `scanf`.

c) Read an integer from keyboard and store that value into variable a & b.

Ans: scanf("%d", &a, &b);

3. What does these code print?

a) printf("\n*\n\n***\n****\n*****");**

```
Output:*
       **
       ***
       ****
       *****
```

b) printf("This is\base");

Output: This is base

c) printf("\n\t\t\t1\n\t\t2\n\t3\n4\n\t5\n\t\t6\t\t\t7"); Output:

```
        1
       2
      3
    4
      5
       6       7
```

Name: Mumtaz Ali

Roll #: 24AI-36

Subject Teacher / LAB Engineer